REMARKS

This is a full and timely response to the outstanding non-final Office Action mailed March 11, 2004. Reconsideration and allowance of the application and presently pending claims 21-43 are respectfully requested.

1. Present Status of Patent Application

Upon entry of the amendments in this response, claims 21-43 remain pending in the present application. More specifically, claims 21, 28, 35 and 38-43 are amended. These amendments are specifically described hereinafter. It is believed that the foregoing amendments add no new matter to the present application.

2. Objection to the Claims

Claims 35, 38 and 40-42 are objected to because of various informalities. The Office Action also suggested amendment to claim 39.

Claims 35 and 38-42 are amended as suggested. Applicant has amended claims 35 and 38-42 as suggested, and thanks the Examiner for the suggested amendments. Accordingly, Applicant requests withdrawal of the objection to these claims.

Applicant notes that the phrase "to test if error recovery is enabled and the subroutine exists" is grammatically equivalent to the phrase "to test if error recovery is enabled and *to test if* the subroutine exists" because of the nature of the word "and" as used in claim 42 as originally filed since the verb "test" is acting upon two conditions; if the "error recovery is enabled" and if the "subroutine exists." However, Applicant agrees that the suggested amendment places claim 42 in better form for allowance, and accordingly, have amended claim 42 as suggested. Furthermore, the amendment to claim 42 has not narrowed the scope of claim 42 in any manner.

Additionally, the amendments to claims 35, 38, 40 and 41 merely correct typographical errors, and do not narrow the scope of these claims in any manner. Furthermore, Applicant believes that the amendments to claims 40 and 41 are broadening amendments.

3. Response to Rejection of Claims 21-43 Under 35 U.S.C. §112, First Paragraph, and Objection to the Drawings under 37 C.F.R. 1.83(a)

In the Office Action, claims 21-43 stand rejected under 35 U.S.C. §112, first and second paragraphs, in the Office Action. Additionally, the drawings are objected to under 37

C.F.R. 1.83(a). Applicant has amended claims 21, 28, 35, 41 and 43 to overcome the rejections. Applicant believes that the amendments to these claims resolve the rejection to the other claims, as these other claims were rejected as being dependent upon a rejected base claim. Additionally, Applicant believes that the amendments to claims 21, 28, 35, 41 and 43 resolve the objection to the drawings. Accordingly, Applicant requests withdrawal of the rejection to these claims and withdrawal of the objection to the drawings.

Applicant notes that the Examiner has interpreted the claims in light of the specification so that examination of the pending claims may proceed on the merits. Applicant thanks the Examiner for interpreting these claims, and appreciates the Examiner's detailed efforts in articulating the basis of Office Action rejections and objections.

4. Response to Rejection of Claims 21-24, 28-30, 34-37 and 41-43 Under 35 U.S.C. §102(b)

In the Office Action claims 21-24, 28-30, 34-37 and 41-43 stand rejected under 35 U.S.C. §102(b) as allegedly being unpatentable over *Agarwal* (U.S. Patent 5,966,541). For a proper rejection of a claim under 35 U.S.C. Section 102, the cited reference must disclose all elements/features/steps of the claim. See, *e.g., E.I. du Pont de Nemours & Co. v. Phillips Petroleum Co.*, 849 F.2d 1430, 7 USPQ2d 1129 (Fed. Cir. 1988).

a. Office Action Interpretation of the Term "Coloring"

The Office Action states that per *Agarwal*, the "identifying/coloring instruction that potentially use dates or selected arguments is interpreted as generating a hidden failure code/flag. That is, a colored instruction is interpreted as a hidden failure code/flag."

Applicant respectfully traverses this interpretation for the reasons below. Furthermore, Applicant asserts that any teachings in *Agarwal* concerning an "identified instruction" or a "colored instruction" (or the like) cannot be used to reject a claim having the feature of a hidden failure code or a hidden failure flag.

A "hidden failure code" or "hidden failure flag" as recited in the claims, and as disclosed in the specification, are ***nouns***. Such a noun (or its equivalent) must be disclosed, taught or suggested in *Agarwal*. Accordingly, only those portions of *Agarwal* that expressly disclose, teach or suggest using a hidden code or a hidden failure flag (or their equivalents) may be used to allege that a recited feature is prior art.

The Examiner is respectfully referred to *Agarwal* at Col. 9, line 66 to Col. 10, line 1, where *Agarwal* discloses that "step 83 is to identify or 'color' the instructions that potentially use dates or selected arguments." Here, *Agarwal* is disclosing that identified/colored instructions are programming code portions (*instructions* which *use* dates or arguments) that are colored.

The Examiner is further directed to *Agarwal* FIG. 7 that "shows a colored graph 400" (FIG. 7 and Col. 10, lines 6-7). "Each ***colored*** ***binary instruction*** is replaced by a set of binary instructions that implement the correct logic" (Col. 10, lines 12-14). Later, *Agarwal* discloses that "instruction I1 at address A2 is assumed to use a date and is therefore shown as colored." Clearly, ***coloring*** as used in *Agarwal* is intended to refer to identified portions of programming code. Programming code is entirely different from a hidden failure flag or a hidden failure code. The Examiner is directed to the Applicant's Specification throughout to appreciate the difference between the coloring of program code sections as used in *Agarwal* and the recitation of a "hidden failure flag" or a "hidden failure code" in the pending claims. Clearly, a "colored instruction" as used in *Agarwal* is not any type of flag. Thus, teachings in *Agarwal* concerning a "colored instruction" or the like cannot be used to reject a claim feature reciting a hidden failure flag or a hidden failure code.

Similarly, ***identifying*** "the instructions that potentially use dates or selected arguments" (Col. 9, line 66 to Col. 10, line 1) is not the same as setting a hidden failure flag or a hidden failure code. That is, identified portions of program code are simply not hidden failure codes or flags. A "hidden failure flag" or "hidden failure code" as recited in the claims are nouns, and such a noun (or its equivalent) must be disclosed, taught or suggested in *Agarwal*. Only those portions of *Agarwal* that expressly disclose, teach or suggest using a hidden failure flag or a hidden failure code (or their equivalents) may be used to allege that a recited feature is prior art.

In the alternative, arguendo, ***even if*** the Office Action somehow construes the *Agarwal* identified instructions (programming code) and/or colored instructions (programming code) to correspond to the Applicant's "hidden failure flag" or "hidden failure code" as recited in the pending claims, there is yet another distinguishing difference. The *Agarwal* identified instructions and/or colored instructions apparently are those portions of programming code corresponding to the failed assertion. That is, during assertion testing, after a determination of a failure of the assertion test is made, *Agarwal* identifies and/or colors the instructions (the portion of programming code) corresponding to the failed assertion. Accordingly, the *Agarwal* identified instructions and/or colored instructions simply are not the same as the Applicant's "hidden failure flag" or "hidden failure code" as recited in the pending claims because the "hidden failure flag" or "hidden failure code" do not identify the actual portion of programming code corresponding to the failed assertion. Rather, the Applicant's "hidden failure flag" or "hidden failure code" merely indicate failure of the assertion test. Thus, for this reason alone, teachings in *Agarwal* concerning an "identified instruction" and/or a "colored instruction" (or the like) cannot be used to reject a claim feature of a "hidden failure flag" or a "hidden failure code" as recited in the pending claims.

b. Independent Claim 21

Applicant respectfully submits that independent claim 21, as amended, is allowable for at least the reason that *Agarwal* does not disclose, teach, or suggest at least the feature of "testing an assertion in the generated *function code*, such that when the assertion is false, the generated *function code* performs the steps of: generating a hidden failure code" and then "***returning to*** the generated ***program code***" as recited in claim 21 (emphasis added).

Applicant respectfully points out that claim 21 recites program code and function code, where it is understood in the art that the function code resides within the program code. Thus, once the hidden failure code is generated (when the assertion is false), there is a return to the generated ***program code*** (thereby leaving the generated ***function code***).

Applicant believes that *Agarwal* does not disclose, teach, or suggest returning to the generated program code after the hidden failure code is generated (when the assertion is tested false within the function code). The Office Action asserts that *Agarwal* discloses that "a preferred embodiment further comprises generating a data flow representation of the binary code, choosing which values or variables to track, and using the data flow representation to track the chosen values or variables, and to further aid in determining where to install the software patches. The control and data flow representations can always be generated from binary code. However, control and data flow representations can also be generated from the source code when the source code is available." (Col. 3, lines 43-52.) This recited portion of *Agarwal* does not disclose, teach, or suggest returning to the generated program code after the hidden failure code is generated (when the assertion is false).

The Office Action then asserts that *Agarwal* discloses that "referring back to FIG. 1B with the aid of the data flow graph, the next step 83 is to identify or 'color' the instructions that potentially use dates or selected arguments. Starting with instructions identified as using dates (or specific arguments) or as being instructions that obtain a date through a program input, data analysis is used to mark or color all the instructions that can be contaminated with a date (or with the specific argument). FIG. 7 shows a colored graph 400 for the case where variable b is a date. In this graph, the hashed nodes N2, N3, N4, N5, and N8 correspond to the instructions that may have to be changed." (Col. 9, line 66 to Col. 10, line 9.) This recited portion of *Agarwal* does not disclose, teach, or suggest ***returning to** the generated **program code*** after the hidden failure code is generated (when the assertion is false).

Applicant notes that *Agarwal* discloses that "the actual rewriting 85, 87, 89 of the binary now takes place. First, the patches are installed 85. Each colored binary instruction is replaced by a set of binary instructions that implement the correct logic. For example, the instruction I3, c=a+b, is replaced in a manner similar to that described earlier. Next, branch and jump instructions are modified if their targets have shifted. This is necessary because when a single instruction is replaced with multiple instructions, the length of that segment of code increases. Thus, the addresses of blocks that follow the lengthened block will all be shifted. Therefore, the branches, procedure calls and jumps that reach a given line of code, or target, through a given old address must also be changed to reflect the new shifted address. The information contained in the control flow graph becomes very useful at this point." (Col. 10, lines 10-25.) However, rewriting code is entirely different from returning to the generated program code after the hidden failure code is generated (when the assertion is false) as recited in claim 21.

Accordingly, for at least this reason alone, *Agarwal* does not anticipate claim 21, and the rejection should be withdrawn.

Additionally, for the reasons detailed above, the identifying or coloring as used in *Agarwal* refers to identified/colored instructions (portions of programming code) that fail assertion testing. The identified/colored instructions are entirely different from the "hidden failure code" recited in claim 21. Thus, teachings in *Agarwal* concerning an identified instruction or a colored instruction cannot be used to properly reject claim 21. Accordingly, and for at least this second reason alone, *Agarwal* does not anticipate claim 21, and the rejection should be withdrawn.

c. Independent Claims 35 and 43

Applicant respectfully submits that independent claim 35, as amended, is allowable for at least the reason that *Agarwal* does not disclose, teach, or suggest at least the feature of "means for testing an assertion in the generated function code, such that when the assertion is false, the generated function code generates a hidden failure code, *returns to* the generated *program code*, and returns the hidden failure code to the generated program code" as recited in claim 35. Similarly, Applicant respectfully submits that independent claim 43, as amended, is allowable for at least the reason that *Agarwal* does not disclose, teach, or suggest at least the feature of "logic that tests an assertion in the generated function code, such that when the assertion is false, the generated function code generates a hidden failure flag, returns to the generated program code, and returns the hidden failure flag to the generated program code" as recited in claim 43.

Applicant respectfully points out that, similar to claim 21 described above, the program code is distinct from the function code (where it is understood that the function code resides within the program code). Thus, once the hidden failure flag or the hidden failure code is generated (when the assertion is tested false within the function code), there is a return to the generated ***program code*** (thereby leaving the generated ***function code***) and a return of the hidden failure code/flag to the generated ***program code***.

Applicant believes that *Agarwal* does not disclose, teach, or suggest returning to the generated program code/flag after the hidden failure code/flag is generated, or returning the hidden failure code/flag to the generated program code (when the assertion is false). The Office Action asserts that *Agarwal* discloses that "a preferred embodiment further comprises generating a data flow representation of the binary code, choosing which values or variables to track, and using the data flow representation to track the chosen values or variables, and to further aid in determining where to install the software patches. The control and data flow representations can always be generated from binary code. However, control and data flow representations can also be generated from the source code when the source code is available." (Col. 3, lines 43-52.) This recited portion of *Agarwal* does not disclose, teach, or suggest returning to the generated program code, or returning the hidden failure code/flag to the generated program code (when the assertion is false).

The Office Action then asserts that *Agarwal* discloses that "referring back to FIG. 1B with the aid of the data flow graph, the next step 83 is to identify or 'color' the instructions that potentially use dates or selected arguments. Starting with instructions identified as using dates (or specific arguments) or as being instructions that obtain a date through a program input, data analysis is used to mark or color all the instructions that can be contaminated with a date (or with the specific argument). FIG. 7 shows a colored graph 400 for the case where variable b is a date. In this graph, the hashed nodes N2, N3, N4, N5, and N8 correspond to the instructions that may have to be changed." (Col. 9, line 66 to Col. 10, line 9.) This recited portion of *Agarwal* does not disclose, teach, or suggest returning to the generated program code, or returning the hidden failure code/flag to the generated program code (when the assertion is false).

Applicant notes that *Agarwal* discloses that "the actual rewriting 85, 87, 89 of the binary now takes place. First, the patches are installed 85. Each colored binary instruction is replaced by a set of binary instructions that implement the correct logic. For example, the instruction I3, c=a+b, is replaced in a manner similar to that described earlier. Next, branch and jump instructions are modified if their targets have shifted. This is necessary because when a single instruction is replaced with multiple instructions, the length of that segment of code increases. Thus, the addresses of blocks that follow the lengthened block will all be shifted. Therefore, the branches, procedure calls and jumps that reach a given line of code, or target, through a given old address must also be changed to reflect the new shifted address. The information contained in the control flow graph becomes very useful at this point." (Col. 10, lines 10-25.) However, rewriting code is entirely different from returning to the generated program code, or returning the hidden failure code/flag to the generated program code, (when the assertion is false) as recited in claims 35 and 43.

Accordingly, *Agarwal* does not anticipate claims 35 and 43, and the rejection should be withdrawn.

Additionally, for the reasons detailed above, the identifying or coloring as used in *Agarwal* refers to identified/colored instructions (portions of programming code) that fail assertion testing. The identified/colored instructions are entirely different from the "hidden failure code" or the "hidden failure flag" recited in claims 35 and 43, respectively. Thus, teachings in *Agarwal* concerning an identified instruction or a colored instruction cannot be used to properly reject claims 35 and 43. Accordingly, and for at least this second reason alone, *Agarwal* does not anticipate claims 35 and 43, and the rejection should be withdrawn.

d. Claims 22-24, 28-30, 34, 36-37 and 41-42

Because independent claim 21 is allowable over the cited art of record, dependent claims 22-24, 28-30 and 34 (which depend from independent claim 21) are allowable as a matter of law for at least the reason that dependent claims 22-24, 28-30 and 34 contain all features/elements/steps of independent claim 21. See, *e.g.*, *In re Fine*, 837 F.2d 1071 (Fed. Cir. 1988). Similarly, because independent claim 35 is allowable over the cited art of record, dependent claims 36-37 and 41-42 (which depend from independent claim 35) are allowable as a matter of law for at least the reason that dependent claims 36-37 and 41-42 contain all features/elements of independent claim 35. Accordingly, the rejection to these claims should be withdrawn.

Additionally, claim 22 is allowable for at least the reason that *Agarwal* does not disclose, teach, or suggest at least the feature of "resetting the hidden failure code" as recited in claim 22. The Office Action alleges that *Agarwal* discloses at "column 10, lines 10-25, a colored instruction is replaced and/or modified, which resets the hidden failure code/flag." As noted above, a colored instruction ***is*** ***<u>not</u>*** a hidden failure code. *Agarwal* discloses that colored instructions are programming code portions (such as dates or arguments) that are colored or identified. Accordingly, replacing and/or modifying the *Agarwal* colored instructions is simply not the same as "resetting the hidden failure code" as recited in claim 22. Thus, *Agarwal* fails to disclose, teach or suggest every element of the Applicant's claimed invention, and the rejection to this claim should be removed.

Furthermore, claims 28 and 41 are allowable for at least the reason that *Agarwal* does not disclose, teach, or suggest at least the feature of "returning to the generated ***<u>program</u>*** ***code*** when the assertion is false" as recited in claims 28 and 41. The Examiner is respectfully referred to the arguments for allowability of claims 21, 35 and 43 which clearly demonstrate that *Agarwal* does not disclose, teach, or suggest at least the feature of "returning to the generated ***<u>program</u>*** ***code*** under any conditions. Thus, *Agarwal* fails to disclose, teach or suggest every element of the Applicant's claimed invention, and the rejection to this claim should be removed.

5. <u>Response to Rejection of Claims 25-27, 31-33 and 38-40 Under 35 U.S.C. §103(a)</u>

In the Office Action claims 25-27, 31 and 38-40 stand rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over *Agarwal*, in view of *Brunmeier* (U.S. Patent 5,511,164). Claims 32 and 33 stand rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over *Agarwal*, in view of *Veldhuizen* (U.S. Patent 5,835,771).

Because independent claim 21 is allowable over the cited art of record, dependent claims 25-27 and 31-33 (which depend from independent claim 21) are allowable as a matter of law for at least the reason that dependent claims 25-27 and 31-33 contain all features/elements/steps of independent claim 21. See, *e.g.*, *In re Fine*, 837 F.2d 1071 (Fed. Cir. 1988). Similarly, because independent claim 35 is allowable over the cited art of record, dependent claims 38-40 (which depend from independent claim 35) are allowable as a matter of law for at least the reason that dependent claims 38-40 contain all features/elements of independent claim 35. Accordingly, the rejection to these claims should be withdrawn.

CONCLUSION

In light of the foregoing amendments and for at least the reasons set forth above, Applicant respectfully submits that all objections and/or rejections have been traversed, rendered moot, and/or accommodated, and that the now pending claims 21-43 are in condition for allowance. Favorable reconsideration and allowance of the present application and all pending claims are hereby courteously requested. If, in the opinion of the Examiner, a telephonic conference would expedite the examination of this matter, the Examiner is invited to call the undersigned agent at (770) 933-9500.

Respectfully submitted,

[illegible]

Raymond W. Armentrout
Reg. No. 45,866